View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by HAL-Univ-Nantes

Motifs récursifs : extraction ascendante hiérarchique d'ensembles d'items ou d'évènements pour le résumé de données transactionnelles ou séquentielles

Julien Blanchard

Université de Nantes & LINA (CNRS UMR6241) équipe COD
Rue Christian Pauc - 44306 Nantes
julien.blanchard@univ-nantes.fr

Résumé. Nous proposons une méthode originale pour extraire un résumé compact, représentatif et intelligible des motifs fréquents dans des données transactionnelles ou séquentielles. Notre approche consiste à extraire un nouveau type de motifs que nous appelons *motifs récursifs*, i.e. des motifs de motifs, à l'aide d'un algorithme hiérarchique agglomératif nommé *RepaMiner*. Nous générons non pas un simple ensemble de motifs mais une véritable structure dérivée de dendrogrammes, le *RPgraph*.

1 Introduction

L'extraction de motifs fréquents est une tâche essentielle en fouille de données. Les motifs permettent de résumer un jeu de données de manière intelligible et peuvent être utilisés pour d'autres tâches comme l'analyse d'association, la classification supervisée associative, ou la classification à base de motifs. Des algorithmes efficaces ont été proposés pour extraire des motifs dans différents types de données comme les données transactionnelles, les séquences d'évènements, et les graphes. Le principal inconvénient des techniques d'extraction de motifs est l'abondance des motifs produits, qui résulte de la nature combinatoire des algorithmes en oeuvre. Différentes solutions ont été proposées face à ce problème, comme l'intégration de contraintes dans les algorithmes (Boulicaut et Jeudy, 2005), le filtrage des motifs par des mesures d'intérêt (Blanchard, 2005; Blanchard et al., 2007), et l'extraction de représentations condensées des motifs fréquents, i.e. un sous-ensemble des motifs qui permet de générer la totalité des motifs de manière exacte ou approchée (Calders et al., 2006). Malgré ces efforts, le problème reste à peine atténué, comme le rappelle l'étude récente de Giacometti et al. (2013).

Dans cet article, nous proposons une méthode originale pour extraire un résumé compact, représentatif et intelligible des motifs fréquents dans des données transactionnelles ou séquentielles (par exemple une ou plusieurs séquences d'évènements, du texte, des séquences biologiques). Ce résumé peut être lu et interprété directement, mais il offre aussi la possibilité de générer de manière approchée l'ensemble des motifs fréquents et d'estimer leur support. Dans le détail, notre approche consiste à extraire un nouveau type de motifs que nous appelons *motifs récursifs*, i.e. des motifs de motifs, à l'aide d'un algorithme hiérarchique agglomératif nommé RepaMiner. La nature hiérarchique de ces motifs nous permet de produire non pas

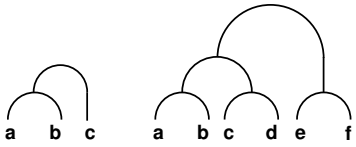


FIG. 1 – *Représentation de deux motifs récursifs par des arbres.*

un simple ensemble de motifs mais une véritable structure, nommée RPgraph, fondée sur les motifs et dérivée de dendrogrammes. Les spécificités de notre approche sont les suivantes :

— L'algorithme RepaMiner est un algorithme dynamique, comme certaines méthodes récentes de fouille de motifs (Vreeken et al., 2011). Les approches dynamiques ont la particularité de modifier le jeu de données analysé à chaque itération en prenant en compte les résultats des itérations précédentes. Pour la fouille de motifs, ces approches restent rares alors qu'elles permettent de diminuer grandement la redondance des résultats produits. Dans le cas de RepaMiner, l'approche dynamique nous permet d'extraire les motifs même avec un seuil de support extrêmement faible.

— L'algorithme RepaMiner peut être vu comme une Classification Ascendante Hiérarchique adaptée aux items (données transactionnelles) et aux évènements (données séquentielles). A l'inverse, les approches habituelles de CAH de variables ne sont pas adaptées aux données binaires déséquilibrées. Adopter une stratégie agglomérative dans RepaMiner nous permet de diminuer l'espace de recherche des motifs d'une taille exponentielle à une taille quadratique (par rapport au nombre d'items ou d'évènements).

2 Extraction ascendante hiérarchique de motifs récursifs

L'algorithme RepaMiner (*Recursive Pattern Miner*) a été conçu à l'origine pour analyser des données séquentielles, l'analyse de données tabulaires étant un simple cas particulier. Cependant, pour présenter notre approche, il est plus clair de nous placer dans le cas classique d'un jeu de données transactionnel. Nous adoptons ce point de vue dans les sections qui suivent. Nous considérons donc un ensemble $\mathcal{I}$ de littéraux nommés items, et un ensemble $\mathcal{T}$ de transactions où chaque transaction t est un ensemble $t \subseteq \mathcal{I}$.

2.1 Motif récursif

Definition 1 (Motif récursif). *Etant donné un ensemble $\mathcal{I}$ d'items, un motif récursif est soit un item de $\mathcal{I}$, soit une paire non ordonnée $\{x, y\}$ où x et y sont des motifs récursifs. x et y sont appelés les parents du motif récursif.*

Example 1. *Soit l'ensemble d'items $\mathcal{I} = \{a, b, c, d, e, f\}$. $m_1 = \{\{a, b\}, c\}$ et $m_2 = \{\{\{a, b\}, \{c, d\}\}, \{e, f\}\}$ sont deux motifs récursifs. Ils sont représentés dans la figure 1.*

Un motif récursif est donc une agrégation de deux éléments, et peut être représenté par un arbre binaire (figure 1). Les motifs récursifs généralisent la notion d'itemset en y introduisant un ordre partiel (les niveaux de l'arbre). Le support dans $\mathcal{T}$ d'un motif récursif m est défini de

manière classique comme étant le nombre de transactions de $\mathcal{T}$ qui contiennent tous les items de m. Les niveaux hiérarchiques des items ne sont donc pas pris en compte dans le calcul du support. Un motif récursif est qualifié de fréquent dans $\mathcal{T}$ si son support est supérieur à un seuil $minsup$ défini par l'utilisateur.

2.2 L'algorithme RepaMiner

L'espace de recherche des motifs récursifs fréquents dans $\mathcal{T}$ est de taille exponentielle par rapport au nombre d'items. Dans RepaMiner, nous préférons adopter la stratégie gloutonne de la CAH afin de réduire l'espace de recherche à une taille quadratique. En partant des motifs récursifs singletons, on agrège à chaque itération les deux motifs récursifs les plus proches au sein d'un nouveau motif récursif, puis on met à jour les données. La similarité entre motifs est évaluée par la surface de leur intersection, définie par *support($m_1 \cap m_2$) $\times$ (nombre d'items dans $m_1 \cap m_2$)*. Cette mesure est utilisée dans les algorithmes de pavage par itemsets (*tiling*). La maximisation de la surface sert aussi bien de mesure de similarité que de critère d'agrégation puisqu'elle s'adapte autant aux singletons qu'aux motifs issus d'agrégations. Parmi une dizaine de mesure étudiées, nous avons constaté empiriquement que maximiser la surface permet à RepaMiner de minimiser l'erreur de restauration des itemsets (Yan et al., 2005).

Algorithm 1 L'algorithme RepaMiner.

Input : ensemble de transactions $\mathcal{T}$, seuil de support minimal $minsup$.
Output : Ensemble des motifs récursifs fréquents $\mathcal{H}$.

1: $\mathcal{H} \leftarrow \emptyset$
2: calculer les items fréquents et leurs supports
3: $\mathcal{C} \leftarrow \{$itemsets fréquents de taille 2$\}$
4: **while** $\mathcal{C} \neq \emptyset$ **do**
5: $\quad m \leftarrow \text{argmax}_{c \in \mathcal{C}}(critereAgregation(c))$
6: $\quad \mathcal{H} \leftarrow \mathcal{H} \cup \{m\}$
7: $\quad$ MiseAJourDonnées($m, minsup, \mathcal{T}$)
8: $\quad \mathcal{C} \leftarrow \{$itemsets fréquents de taille 2$\}$
9: **end while**
10: **return** $\mathcal{H}$

RepaMiner n'utilise que des méthodes classiques de calcul d'itemsets (motifs ensemblistes) de taille 2. C'est la propriété dynamique de notre approche (création de nouveaux items à chaque itération) qui permet de construire des hiérarchies d'items à l'aide de méthodes d'extraction d'itemsets de taille 2. Le pseudo-code est résumé dans l'algorithme 1. $\mathcal{C}$ est l'ensemble des motifs récursifs candidats, parmi lesquels le meilleur (au sens du critère d'agrégation) sera choisi. A la ligne 3, $\mathcal{C}$ est initialisé à l'ensemble des itemsets de taille 2 fréquents dans $\mathcal{T}$, accompagnés de leur support. A la ligne 5, on identifie le meilleur motif candidat m au sens du critère d'agrégation. La procédure *MiseAJourDonnées()* de la ligne 7 modifie le jeu de données $\mathcal{T}$. Un nouvel item est créé pour coder les occurrences du motif m, et toutes les occurrences des parents de m qui participent au support de m sont supprimées. L'idée est de ne pas perdre l'information représentée par les 1 dans les données, en la répartissant parmi les trois items. La procédure s'achève en supprimant les items qui sont devenus non fréquents

(a) Dendrogrammes (b) RPgraph

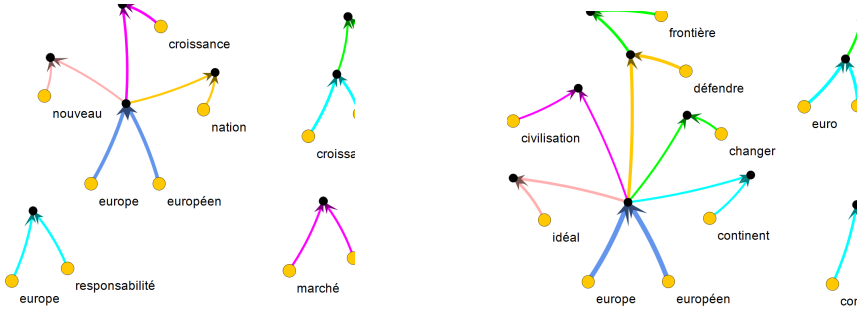
FIG. 2 – *Deux représentations équivalentes des motifs récursifs. Le RPgraph se lit ainsi :* a *s'agrège avec* b*, puis avec* c ***OU*** *avec* d *puis* e.

suite à la mise à jour des données. A la ligne 8, on calcule les itemsets de taille 2 fréquents dans $\mathcal{T}$ en prenant en compte les mises à jour effectuées sur $\mathcal{T}$ à la ligne 7. Ces itemsets constituent le nouvel ensemble $\mathcal{C}$ de candidats. Lorsqu'il n'y a plus aucun motif fréquent candidat au titre de meilleure agrégation, l'algorithme retourne l'ensemble $\mathcal{H}$ des motifs récursifs qui ont été extraits, i.e. l'ensemble des agrégations qui ont été réalisées. Chaque motif de $\mathcal{H}$ est fréquent (par construction), et est accompagné de sa valeur de support au moment de l'agrégation.

Résultats de l'algorithme. RepaMiner produit l'ensemble $\mathcal{H}$ des motifs récursifs qui ont été générés à chaque itération. Chaque motif est accompagné de la valeur de support qu'il présentait au moment de sa création. Pour visualiser l'ensemble $\mathcal{H}$, nous représentons chaque motif récursif maximal [1] par un dendrogramme indicé par le complément à 1 du support. Ce choix est justifié par le fait que le complément du support est une ultramétrique sur l'ensemble des items produits par RepaMiner. Il est à noter que $\mathcal{H}$ ne constitue une hiérarchie unique et complète que dans le cas où le seuil de support $minsup$ est nul. Avec RepaMiner, dans le cas général, le seuil $minsup$ impose une coupure dans le dendrogramme. On obtient alors plusieurs hiérarchies déconnectées, chacune ayant à son sommet un motif récursif maximal.

Visualisation du flux agglomératif à l'aide d'un RPgraph. Nous tirons profit des recouvrements entre les motifs récursifs maximaux de $\mathcal{H}$ pour construire une représentation plus synthétique, que nous nommons RPgraph (*Recursive Pattern graph*). Un RPgraph peut être vu comme une vue "de dessus" des dendrogrammes représentant chaque motif maximal. Cette vue nous prive de la hauteur d'agrégation, i.e. le support du motif, mais permet de bénéficier de davantage d'espace pour disperser les structures dans le plan. Nous profitons de cet espace pour réunir les dendrogrammes qui ont des intersections communes. Ce processus est illustré en figure 2.a. Au final, nous obtenons un graphe dont les noeuds représentent les motifs récursifs (soit un item, soit une agrégation), et les arcs relient les motifs agrégés. Par exemple, un motif récursif $m = \{x, y\}$ est représenté par un noeud m qui est relié à x et y par deux arcs (x, m) et (y, m). Les arcs sont orientés vers m pour montrer le "flux agglomératif" découvert dans les données. Le support du motif récursif m est représenté par la largeur des deux arcs. Les couleurs des arcs servent uniquement à repérer les deux branches (x, m) et (y, m) d'une même agrégation, en leur donnant la même couleur.

1. Un motif récursif $m_1 \in \mathcal{H}$ est dit maximal ssi il n'existe aucun motif m_2 et M de $\mathcal{H}$ tels que $M = \{m_1, m_2\}$.



(a) François Hollande (b) Nicolas Sarkozy

FIG. 3 – *Résumé de discours politiques au moyen de RPgraphs.*

Cas général : les données séquentielles. L'algorithme RepaMiner intègre un pré-traitement pour pouvoir analyser des données séquentielles, i.e. une ou plusieurs séquences continue(s) ou discrète(s) d'évènements instantanés ou ponctuels. Ce pré-traitement est fondé sur le cadre formel de la la découverte d'épisodes fréquents, initié par les travaux de Mannila et al. (1997). Nous nous inspirons de la méthode Winepi, qui consiste à appliquer une fenêtre glissante sur la séquence pour produire un ensemble de sous-séquences légèrement décalées et de longueurs identiques. Dans RepaMiner, chaque sous-séquence est transformée en transaction, i.e. l'ordre entre évènements est supprimé. L'ensemble de transactions obtenu devient l'ensemble $\mathcal{T}$ en entrée de l'algorithme 1, et l'ensemble des types d'évènements présents dans les séquences constitue l'ensemble $\mathcal{I}$ des items. RepaMiner extrait donc des motifs non ordonnés dans des données ordonnées (ce que Mannila, Toivonen, et Verkamo appellent des épisodes parallèles).

3 Illustration sur données réelles : 50 discours politiques

Nous illustrons notre démarche en résumant 50 discours [2] des candidats du second tour de l'élection présidentielle 2012 à l'aide de RepaMiner. Chaque discours a été lemmatisé et confronté à une liste de *stop-words* avant de constituer une séquence de lemmes (les lemmes jouent le rôle des items). Les lemmes les plus rares (moins de 8 occurrences) ont également été supprimés. Nous obtenons pour François Hollande 21 séquences de 1133 lemmes en moyenne parmi un vocabulaire de 875 lemmes différents, et pour Nicolas Sarkozy 29 séquences de 913 lemmes en moyenne parmi 914 lemmes différents. En appliquant RepaMiner avec un seuil de support $minsup = 0.5\%$ et une taille de fenêtre $\omega = 30$ lemmes consécutifs, nous obtenons 357 motifs récursifs fréquents maximaux pour François Hollande et 303 pour Nicolas Sarkozy. Les figures 3.a et 3.b sont issues des deux RPgraphs générés et montrent les composantes connexes qui concernent le thème de l'Europe. Chaque agrégation entre deux lemmes peut être interprétée ainsi : les deux lemmes apparaissent fréquemment dans les mêmes sous-séquences de longueur 30. On peut percevoir dans ces résumés des axes de communication et d'analyse

2. www.lemonde.fr/election-presidentielle-2012/visuel/2012/03/15/explorez-les-discours-des-candidats-a-la-presidentielle-2012_1669414_1471069.html, 50 discours énoncés entre le 19/02 et le 17/04/2013.

différents de la part des deux candidats. Dans notre implémentation de RepaMiner, un slider permet de rejouer la construction du RPgraph pour visualiser le flux agglomératif.

4 Conclusion

Nous avons proposé une méthode originale pour extraire un résumé compact, représentatif et intelligible des motifs fréquents dans des données transactionnelles ou séquentielles. Cette méthode repose sur la notion de motif récursif, et sur l'algorithme dynamique RepaMiner qui permet de réaliser l'extraction ascendante hiérarchique de ces motifs. Nous générons une véritable structure nommée RPgraph qui permet de visualiser un "flux agglomératif" dans les données. Au final, RepaMiner peut être vu comme une Classification Ascendante Hiérarchique adaptée aux items (données transactionnelles) et aux évènements (données séquentielles). Ce travail se poursuit par des expérimentations qui montrent que le résumé généré est représentatif puisqu'il permet d'estimer précisément l'ensemble des itemsets ou épisodes fréquents.

Références

Blanchard, J. (2005). *Un système de visualisation pour l'extraction, l'évaluation, et l'exploration interactives des règles d'association*. Ph. D. thesis, Université de Nantes.

Blanchard, J., F. Guillet, et R. Gras (2007). On the discovery of significant temporal rules. In *Proceedings of the IEEE Conference SMC'2007*, pp. 443–450. IEEE Society Press.

Boulicaut, J.-F. et B. Jeudy (2005). Constraint-based data mining. In O. Maimon et L. Rokach (Eds.), *Data Mining and Knowledge Discovery Handbook*, pp. 399–416. Springer.

Calders, T., C. Rigotti, et J.-F. Boulicaut (2006). A survey on condensed representations for frequent sets. In *Constraint-Based Mining and Inductive Databases*, Volume 3848 of *Lecture Notes in Computer Science*, pp. 64–80. Springer.

Giacometti, A., D. Haoyuan Li, et A. Soulet (2013). 20 ans de découverte de motifs : une étude bibliographique quantitative. In *Actes EGC'2013*, pp. 133–144. Hermann-Éditions.

Mannila, H., H. Toivonen, et A. I. Verkamo (1997). Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery 1*(3), 259–289.

Vreeken, J., M. Leeuwen, et A. Siebes (2011). Krimp : mining itemsets that compress. *Data Mining and Knowledge Discovery 23*(1), 169–214.

Yan, X., H. Cheng, J. Han, et D. Xin (2005). Summarizing itemset patterns : a profile-based approach. In *Proc. of the ACM SIGKDD conference KDD'05*, pp. 314–323. ACM.

Summary

We propose an original method to mine a compact, representative and intelligible summary of the frequent patterns in transactional or sequential data. Our method consists in mining a new kind of patterns that we call *Recursive Patterns*, i.e. patterns of patterns, with an agglomerative hierarchical algorithm named *RepaMiner*. We generate not only a set of patterns but a true structure, named *RPgraph*, derived from dendrograms.